**Task 2 Documentation: One-Click Installer for Python Application**

**1. Steps Taken**

**Step 1: Setup Python Environment**

* First, I ensured that Python was installed on the machine (version 3.10 in this case).
* I installed all required libraries using pip install. Key dependencies included PyInstaller for creating the standalone executable using the command:

**“pip install pyinstaller”**

**Step 2: Create the .spec File**

* After testing the basic PyInstaller functionality using the command pyinstaller install.py, I realized the need to customize the packaging process by creating a .spec file.
* The .spec file was tailored to include additional dependencies, such as local modules that may not be detected automatically by PyInstaller. For instance, I added the local-deps/ to the data section in the .spec file to ensure that required files, like .whl dependencies, are bundled with the .exe.

**Step 3: Adjusting the .spec File**

The original .spec file was modified to look like this:

**a = Analysis(**

**['install.py'],**

**pathex=[], # Add the path to your project directory if needed**

**binaries=[],**

**datas=[('local-deps/\*', 'local-deps')], # Include the local-deps folder**

**hiddenimports=[],**

**hookspath=[],**

**hooksconfig={},**

**runtime\_hooks=[],**

**excludes=[],**

**noarchive=False,**

**optimize=0,**

**)**

**Step 4: Build the Executable**

* Once the .spec file was set up, I ran PyInstaller with the command:

**“pyinstaller install.spec”**

* This generated the standalone executable install.exe located in the dist directory.

**Step 5: Test the Executable**

* I tested the generated install.exe on my machine. Upon double-clicking, the installer ran as expected, but I encountered some issues with missing dependencies.

**2. Challenges Faced**

* **Missing Dependencies**
* During the testing phase, I noticed that some dependencies, which were required for the application, weren’t bundled into the .exe file.
* The local-deps folder containing .whl files (such as numpy.whl, pandas.whl, etc.) wasn’t being correctly recognized as part of the executable.
* **Solution:** I modified the .spec file to explicitly include these files by referring to the local-deps folder under the data section.

**3. Conclusion**

This task successfully resulted in the creation of a one-click installer for a Python application. Despite initial challenges with missing dependencies, I was able to resolve them by modifying the .spec file and ensuring all necessary .whl files were bundled. The final executable can now be distributed to users without the need for them to install Python or manually manage dependencies